We used Selenium-request package with Firefox and urllib to access 500 websites to get the response codes. For each test we used two different permutations for tests. For the first permutation, we randomized the sequence of using Selenium and urllib. We repeated this randomized pair 3 times. The second one, we randomized the 6 accesses, 3 for urllib and 3 for Selenium. For each of the tests mentioned below we run it in both of these two formats to see if we find any major difference.

Selenium/Urllib
We used the method mentioned above to get the response code from 500 websites with 6 accesses in each test. Approximately, Selenium got 1350 successful response codes(200), urllib got 1250 successful response codes, both with the time out of 5 seconds.

Selenium/Urllib with changed user agent
We changed the user agent of urllib to match the user agent of Firefox(since we were using Selenium with firefox.) Urllib then could get almost the same amount(around 1350) of 200 response code as Selenium. We proceeded this test by changing the user agent to other strings. When we changed the user agent of Urllib to only “Mozzila/5.0,” urllib got around 1300 successful response codes. After this test we tried “Mozzila/5.01234,” “Mozzila/5.0” followed with 20 random characters, and “Mozzila/5.0” followed with 50 random characters, all 3 tests got less then 10 successful responses than Selenium. We ran the above 3 tests with time out limit of 15 seconds instead of 5 second. The amount of successfully responses increased by about 20 for both browsers.

Chrome/FireFox
We tried to use Chrome and Firefox with Selenium to see if one got better performance. We used the same kind of permutations when we tested with Selenium and urllib. With the time limit of 15 seconds, Chrome got 350 less successful response codes from Selenium. After we increased the time limit to 30 seconds, Chrome got almost as many successful response codes as Firefox. The difference is caused by Firefox finishes the session before the window fully loaded. Chrome finishes the session only when the window fully loads.